



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/647,058	08/21/2003	J. Patrick Thompson	MSFT-1748/302722.01	1588

41505 7590 08/17/2010  
WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)  
CIRA CENTRE, 12TH FLOOR  
2929 ARCH STREET  
PHILADELPHIA, PA 19104-2891

EXAMINER
----------

PHAM, MICHAEL

ART UNIT	PAPER NUMBER
----------	--------------

2167

MAIL DATE	DELIVERY MODE
-----------	---------------

08/17/2010

PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

### Office Action Summary

**Application No.**

10/647,058

**Applicant(s)**

THOMPSON ET AL.

**Examiner**

MICHAEL PHAM

**Art Unit**

2167

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 16 June 2010.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-11, 13-20, 23 and 37-48 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-11, 13-20, 23, and 37-48 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB06)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_

**DETAILED ACTION**

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 6/16/2010 has been entered.

***Claim Status***

2. Claims 1-11, 13-20, 23, and 37-48 are pending and examined.

***Specification***

3. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

***Claim Rejections - 35 USC § 112***

4. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

5. Claims 1, 37, and 43 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which

was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. Applicant's cite paragraphs 0777-0794 of the application as published in order to support the new limitations of claim 1 "the instructions for the database management program further including instructions for receiving receive read/write requests from the user mode applications for Items via one or more functions of an operating system application program interface; and the instructions for the database management program further including instructions for deserializing files storing the file data for the Items into Items and return the Items to the user mode applications" of which are similarly recited in claims 37 and 43. No where does it state "receiving receive read/write requests from the user mode applications for Items via one or more functions of an operating system application program interface " and "deserializing files storing the file data for the Items into Items". There's no support for the claimed limitations. At best it appears the closest thing to being close to what "deserializing files storing the file data for the Items into Items" states is on 0784, which states "deserializes data read from the store into objects."; however it makes no sense as to what is considered storing the file data for the Items into Items for example. Furthermore, in regards to "receiving receive read/write requests from the user mode applications for Items via one or more functions of an operating system application program interface ", it is not clear where this is even disclosed in the cited paragraphs.

***Claim Rejections - 35 USC § 101***

6. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

7. Regarding claims 43-48 these claims recite a 'computer readable storage medium' and 'processor'. In the absence of any modifying disclosure of this limitation in the specification, the examiner interprets the terms 'computer readable storage medium' and the term 'processor' as limited to statutory embodiments only, such that the claims fall within a statutory class of invention as required under the terms of 35 U.S.C. 101.

8. Regarding claims 1-11, 13-20, 23 these claims recite a 'computer readable storage medium' and 'processor'. In the absence of any modifying disclosure of this limitation in the specification, the examiner interprets the terms 'computer readable storage medium', and the term 'processor' as limited to statutory embodiments only, such that the claims fall within a statutory class of invention as required under the terms of 35 U.S.C. 101.

***Claim Rejections - 35 USC § 103***

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. **Claims 1, 37, and 43 rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6018342 by David Bristor (hereafter Bristor) further in view of U.S. Patent**

**Application Publication 20030009685 by Choo et. al. (hereafter Choo) and U.S. Patent 7158962 by Kenneth Nelson (hereafter Nelson).**

**Claim 1**

Bristor discloses the following claimed limitations:

“a processor coupled to a computer readable storage medium, the computer readable storage medium including” [figure 6, computer system]

“the file system configured to store file data as filestreams and the database management program is configured to generate Items from the file data and expose the Items to a shell of the operating system, the Items constituting discrete storable units of information;”[Figure 1, mystuff; mystuff.txt, mystuff.dat, mystuff.c. Accordingly, the file system (mystuff e.g. file system containing files) configured to store file data as filestreams (mystuff.txt, mystuff.dat, mystuff.c) and the database management program (command) is configured to generate Items (Figure 1B, 112L) from the file data (mystuff) and expose (figure 1B) the Items (112 commands) to a shell of the operating system (col. 8 lines 41-50, processed as if user had entered the command), the Items constituting discrete storable units of information (figure 1B element 112L).]

“the instructions for the database management program further including instructions for generating a plurality of Item Folders that constitute an organizational structure for said Items;”[Figure 1B 108. Accordingly, the instructions for the database management program (figure 1B, command prompt 102) further including instructions for generating a plurality of Item Folders (figure 1B element 108) that constitute an organizational structure for said Items (figure 1B element 108 M)]

“and each Item Folder includes membership information identifying any relationships with Items;”[Figure 1B element 108; col. 8 lines 4-8, in accordance with the present invention, an organization scheme which is symbolic and with which users are already familiar is used in a novel manner to categorize previously generated user data. Accordingly, each Item folder (figure 1B element 108) includes membership information (fig. 1B, M) identifying any relationships with Items (fig. 1B, 112)]

“the instructions for the database management program including instructions for generating a plurality of Categories that constitute an additional organizational structure for said Items, at least one of said Items belonging to at least one of the Categories”[ Abstract lines 3-8, regeneration in a history database in one of two or more categories associated with two or more respective component symbols of the user data. For example, in one embodiment, user data includes alphabetic symbols and a respective category is formed for each letter of the alphabet. Accordingly, the instructions for the database management program including instructions for generating a plurality of Categories (Categories) that constitute an additional organizational structure (respective category is formed for each letter of the alphabet) for said Items (user data/commands 112), at least one of said Items belonging to at least one of the Categories(figure 1B element 112L).

“wherein each Item in a specific Category includes a common attribute that is described for that specific Category”[figure 1C elements 112MA-MC, 108M]

Bristol does not explicitly disclose

“the computer readable storage medium including instructions for an operating system including instructions for a database management program, the instructions for the database management program configured to control user mode application access to a file system”

“the instructions for the database management program further including instructions for receiving receive read/write requests from the user mode applications for Items via one or more functions of an operating system application program interface; and”

“the instructions for the database management program further including instructions for deserializing files storing the file data for the Items into Items and return the Items to the user mode applications.”

On the other hand, Choo discloses the following claimed limitations:

“the computer readable storage medium including instructions for an operating system including instructions for a database management program, the instructions for the database management program configured to control user mode application access to a file system,”[ See figure 2-3. Accordingly, the computer readable storage medium (figure 3) including instructions for an operating system (figure 2 element 200, system) including instructions for a database management program (figure 2 element 203, rules database), the instructions for the database management program (figure 2 element 203, rules database)configured to control (0014 line 15-16, determine whether the compartment associated with process 201 is permitted access) user mode application (figure 2 element 201, process executes code in user-space) access to a file system (0014 line 16, permitted access to the particular resource)]



“the instructions for the database management program further including instructions for receiving receive read/write requests from the user mode applications for Items via one or more functions of an operating system application program interface; and”[ Figure 2 and 3.

Paragraphs 0014-0016. Accordingly, the instructions for the database management program (figure 2 element 203, rules database) further including instructions for receiving receive read/write requests (0016, command line utilities may further include commands to create, delete, and/or modify the rules stored in the rules database 316) from the user mode applications (0016 line 1, processes) for Items (0016 line 8, resources) via one or more functions of an operating system application program interface (0016 lines 2-3, command line utilities)]

“the instructions for the database management program further including instructions for deserializing files storing the file data for the Items into Items and return the Items to the user mode applications.”[ Figure 2 and 3. Paragraphs 0014-0016. Accordingly, the instructions for the database management program (figure 3 rules database) further including instructions (various components and modules for the various types of resources)for deserializing files (limit access to system resources) storing the file data (types of resources) for the Items (resources)into Items (access system resources)and return the Items to the user mode applications (process is permitted access to the particular resource)]

Both Bristor and Choo disclose the systems that manipulate file systems. They are therefore within applicant’s same field of endeavor and are therefore analogous. Bristor provides a database management program that allows access to file data, see figure 1A and 1B. Choo discloses a rules database management program within a kernel of the operating system, see figure 2. Choo discloses that the rules database management program provides for authorized

access to particular file resources, see abstract. It would have therefore been obvious to a person of an ordinary skill in the art at the time the invention was made to have provided the disclosure of Choo above to the disclosure of Bristol for the purpose of defining authorization to file resources. Doing so would improve upon Bristol's system by providing protection to files in a file system.

The combination of Bristol and Choo do not explicitly disclose

"wherein the Item Folders themselves are Items"; and

"Categories are themselves Items";

On the other hand, Nelson discloses

"wherein the Item Folders are themselves Items" [col. 3 lines 59-60, items can be for example folder or a document], and

"Categories are themselves items" [col. 1 lines 38-42, each item may exist as the source and/or target]

Bristol, Choo, and Nelson are all within applicant's same field of endeavor as they are all directed to filing systems. Nelson discloses that the Categories can be items as seen in col. 1 lines 38-42, and further discloses that the Item folders are themselves Items as can be seen in col. 3 lines 59-60. The combination of Bristol and Choo disclose in Bristol that items are organized, see abstract lines 3-8. It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to have applied Nelson's disclosure above to the combination of Bristol and Choo for the purpose of further categorizing and organizing items.

**Claim 37:**

Bristor discloses the following claimed limitations:

“storing, by the file system, file data;”[figure 1, mystuff; mystuff.txt, mystuff.dat, mystuff.c. Accordingly, the storing, by the file system (mystuff, e.g. file system containing files), file data (mystuff.txt, mystuff.dat, mystuff.c).]

“generating, by the database management program, Items from the file data, wherein each Item of said plurality of items constitutes a discrete unit of information;”[ Figure 1. Accordingly, generating, by the database management program (figure 1B), Items from the file data (figure 1B element 112L), wherein each Item of said plurality of items constitutes a discrete unit of information (figure 1B element 112L)]

“generating, by the database management program from the file data, Item Folders, the Item Folders constituting an organizational structure for said Items” [Figure 1. Accordingly, generating, by the database management program (figure 1B) from the file data (figure 1B, mystuff), Item Folders (figure 1B element 108), the Item Folders constituting an organizational structure (figure 1B element 108) for said Items (figure 112L)]

“exposing, by the database management program, the Items and the Item Folders to a shell of the operating system.”[ Accordingly, exposing (col. 8 line 44, to retrieve and re-enter command), by the database management program (figure 1B 102), the Items (figure 1B element 112L) and the Item Folders (figure 1B 108) to a shell of the operating system (col. 10 line 64, c-shell)]

Bristor does not explicitly disclose:

“executing an operating system including a database management program, the database management program configured to control user mode application access to a file system;”

“receiving, by the database management program, read/write requests from the user mode applications for Items via one or more functions of an operating system application program interface;”

“deserializing, by the database management program, files storing the file data for the Items into Items.”

On the other hand, Choo discloses

“executing an operating system including a database management program, the database management program configured to control user mode application access to a file system;”[ See figure 2-3, See figure 2-3. Accordingly, executing an operating system (figure 2 element 200, system) including a database management program (figure 2 element 203, rules database), the database management program (figure 2 element 203, rules database)configured to control (0014 line 15-16, determine whether the compartment associated with process 201 is permitted access) user mode application (figure 2 element 201, process executes code in user-space) access to a file system (0014 line 16, permitted access to the particular resource)]

“receiving, by the database management program, read/write requests from the user mode applications for Items via one or more functions of an operating system application program interface;”[ Figure 2 and 3. Paragraphs 0014-0016. Accordingly, receiving, by the database management program (figure 2 element 203, rules database), read/write requests (0016, command line utilities may further include commands to create, delete, and/or modify the rules stored in the rules database 316) from the user mode applications (0016 line 1, processes) for Items (0016 line 8, resources) via one or more functions of an operating system application program interface (0016 lines 2-3, command line utilities)]

“deserializing, by the database management program, files storing the file data for the Items into Items.”[ Figure 2 and 3. Paragraphs 0014-0016. Accordingly, deserializing (limit access to), by the database management program (figure 3 rules database), files (system resources) storing the file data (types of resources) for the Items (resources) into Items (access system resources)]

Both Bristor and Choo disclose the systems that manipulate file systems. They are therefore within applicant's same field of endeavor and are therefore analogous. Bristor provides a database management program that allows access to file data, see figure 1A and 1B. Choo discloses a rules database management program within a kernel of the operating system, see figure 2. Choo discloses that the rules database management program provides for authorized access to particular file resources, see abstract. It would have therefore been obvious to a person of an ordinary skill in the art at the time the invention was made to have provided the disclosure of Choo above to the disclosure of Bristor for the purpose of defining authorization to file resources. Doing so would improve upon Bristor's system by providing protection to files in a file system.

On the other hand, Nelson discloses

“wherein the Item Folders are a type of Item” [col. 3 lines 59-60, items can be for example folder or a document]

Bristor, Choo, and Nelson are all within applicant's same field of endeavor as they are all directed to filing systems. They are all therefore analogous and relevant. Nelson discloses that the Categories can be items as seen in col. 1 lines 38-42, and further discloses that the Item folders are themselves Items as can be seen in col. 3 lines 59-60. The combination of Bristor and

Choo disclose in Bristol that items are organized, see abstract lines 3-8. It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to have applied Nelson's disclosure above to the combination of Bristol and Choo for the purpose of further categorizing and organizing items.

**Claim 43:**

Bristol discloses the following claimed limitations:

“instructions for the file system configured to store file data;” [figure 1, mystuff; mystuff.txt, mystuff.dat, mystuff.c. Accordingly, instructions for the file system (mystuff, e.g. file system containing files) configured to store file data(mystuff.txt, mystuff.dat, mystuff.c)]

“the instructions for the database management program further including instructions for generating a plurality of Items, said Item comprising a discrete unit of information from the file data;” [ Figure 1. Accordingly, instructions for the database management program (figure 1B) further including instructions for generating a plurality of Items (figure 1B element 112L; figure 1C 112MA-MC), said Item comprising a discrete unit of information from the file data]

“instructions for the database management program further including instructions for generating a plurality of Item Folders from the file data, the Item Folders including membership information identifying any relationships with Items;”[ Accordingly, instructions for the database management program further including instructions for generating (figure 1B) a plurality of Item Folders (figure 1B element 108) from the file data (figure 1B mystuff), the Item Folders (figure 1B element 108) including membership information identifying any relationships (figure 1B element 108) with Items (figure 1B element 112)]

“instructions for the database management program further including instructions for generating a plurality of Categories from the file data, the Categories constituting an organizational structure for said Items;”[ Figure 1B. Accordingly, instructions for the database management program further including instructions for generating (figure 1B element 102) a plurality of Categories (figure 1B, A-Z) from the file data (figure 1B is mystuff), the Categories (figure 1B, A-Z) constituting an organizational structure (alphabetical) for said Items (figure 1B 112)]

“instructions for the database management program further including instructions for exposing the Items to a shell of the operating system.”[figure 1B. Accordingly, instructions for exposing (col. 8 line 44 to retrieve and re-enter command), by the database management program (figure 1B element 102), the Items to a shell of the operating system (col. 10 line 64, c-shell)]

Bristor does not explicitly disclose

“instructions for an operating system database management program, the instructions for the database management program configured to control user mode application access to a file system;”

On the other hand, Choo discloses

“instructions for an operating system database management program, the instructions for the database management program configured to control user mode application access to a file system;”[ See figure 2-3, Accordingly, instructions for an operating system (figure 2 element 200) including a kernel mode (figure 2, kernel) and a user mode (figure 2, user space), the kernel mode (figure 2 kernel) of the operating system (figure 2 200) including instructions for a database management program (figure 2 element 203), the instructions for the database

management program (figure 2 element 203) encapsulating instructions (rules) for a file system (file resources)]

Both Bristor and Choo disclose the systems that manipulate file systems. They are therefore within applicant's same field of endeavor and are therefore analogous. Bristor provides a database management program that allows access to file data, see figure 1A and 1B. Choo discloses a rules database management program within a kernel of the operating system, see figure 2. Choo discloses that the rules database management program provides for authorized access to particular file resources, see abstract. It would have therefore been obvious to a person of an ordinary skill in the art at the time the invention was made to have provided the disclosure of Choo above to the disclosure of Bristor for the purpose of defining authorization to file resources. Doing so would improve upon Bristor's system by providing protection to files in a file system.

The combination of Bristor and Choo do not explicitly disclose

"the Item Folders are themselves a type of Item"

On the other hand, Nelson discloses

"and the Item Folders are themselves a type of Item" [col. 3 lines 59-60, items can be for example folder or a document]

Bristor, Choo, and Nelson are all within applicant's same field of endeavor as they are all directed to filing systems. They are all therefore analogous and relevant. Nelson discloses that the Categories can be items as seen in col. 1 lines 38-42, and further discloses that the Item folders are themselves Items as can be seen in col. 3 lines 59-60. The combination of Bristor and Choo disclose in Bristor that items are organized, see abstract lines 3-8. It would have been



obvious to a person of an ordinary skill in the art at the time the invention was made to have applied Nelson's disclosure above to the combination of Bristor and Choo for the purpose of further categorizing and organizing items.

**11. Claims 2-3, 5, 7-8, 10, 38, 39, 41, 44, 45, and 47 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6018342 by David Bristor (hereafter Bristor) further in view of U.S. Patent Application Publication 20030009685 by Choo et. al. (hereafter Choo), U.S. Patent 7158962 by Kenneth Nelson (hereafter Nelson), and U.S. Patent Application Publication 2004/0199521 by Anglin et. al (hereafter Anglin).**

**Claim 2 :**

Bristor, Choo, and Nelson do not explicitly disclose "wherein an Item is a member of an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said Item." On the other hand, Anglin discloses "wherein an Item is a member of an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said Item." (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.) Bristor, Choo, Nelson, and Anglin are all directed to the same field of endeavor as applicant's invention, as they are systems of file organization and

management. It would have been obvious to a person of an ordinary skill in the art at the time the invention was disclosed to have applied Anglin's disclosure above to the combination of Bristor, Choo, and Nelson for the purpose of providing a removal method for the item.

**Claim 3 :**

Bristor, Choo, and Nelson do not explicitly disclose "wherein an Item is automatically deleted when it no longer belongs to any Item Folder." On the other hand, Anglin discloses "wherein an Item is automatically deleted when it no longer belongs to any Item Folder." (See page 3, paragraph [0029] "If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.") Bristor, Choo, Nelson, and Anglin are all directed to the same field of endeavor as applicant's invention, as they are systems of file organization and management. It would have been obvious to a person of an ordinary skill in the art at the time the invention was disclosed to have applied Anglin's disclosure above to the combination of Bristor, Choo, and Nelson for the purpose of providing a removal method for the item.

**Claim 5 :**

The combination of Bristor, Choo, and Nelson do not explicitly disclose "wherein said Item is automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted." On the other hand, Anglin discloses "wherein said Item is automatically deleted when it

is a member of only one Item Folder and said Item Folder is deleted.” (See page 3, paragraph [0028] “The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i.” This follows the request to delete the “group leader” which represents the group ID of the “storage group” or the “Item Folder” as in referred to in the claim.) Bristor, Choo, Nelson, and Anglin are all directed to the same field of endeavor as applicant’s invention, as they are systems of file organization and management. It would have been obvious to a person of an ordinary skill in the art at the time the invention was disclosed to have applied Anglin’s disclosure above to the combination of Bristor, Choo, and Nelson for the purpose of providing a removal method for the item.

**Claim 7 :**

Bristor, Choo, and Nelson do not explicitly disclose “wherein each Item is a member of at least one Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of an Item.” Anglin discloses “wherein each Item is a member of at least one Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of an Item.” (See page 3, paragraph [0028] “The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i.” This follows the request to delete the “group leader” which represents the group ID of the “storage group” or the “Item Folder” as in referred to in the claim.) Bristor, Choo, Nelson, and Anglin are all directed to the

same field of endeavor as applicant's invention, as they are systems of file organization and management. It would have been obvious to a person of an ordinary skill in the art at the time the invention was disclosed to have applied Anglin's disclosure above to the combination of Bristol, Choo, and Nelson for the purpose of providing a removal method for the item.

**Claim 8 :**

Bristol, Choo, and Nelson do not explicitly disclose "wherein each said Item is itself automatically deleted when it no longer belongs to any Item Folder." Anglin discloses "wherein each said Item is itself automatically deleted when it no longer belongs to any Item Folder." (See page 3, paragraph [0029] "If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.") Bristol, Choo, Nelson, and Anglin are all directed to the same field of endeavor as applicant's invention, as they are systems of file organization and management. It would have been obvious to a person of an ordinary skill in the art at the time the invention was disclosed to have applied Anglin's disclosure above to the combination of Bristol, Choo, and Nelson for the purpose of providing a removal method for the item.

**Claim 10 :**

Bristol, Choo, and Nelson do not explicitly disclose "wherein each said Item is itself automatically deleted when it is a member of only one Item Folder and said Item Folder is

deleted.” Anglin discloses “wherein each said Item is itself automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted.” (See page 3, paragraph [0028] “The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i.” This follows the request to delete the “group leader” which represents the group ID of the “storage group” or the “Item Folder” as in referred to in the claim.) Bristol, Choo, Nelson, and Anglin are all directed to the same field of endeavor as applicant’s invention, as they are systems of file organization and management. It would have been obvious to a person of an ordinary skill in the art at the time the invention was disclosed to have applied Anglin’s disclosure above to the combination of Bristol, Choo, and Nelson for the purpose of providing a removal method for the item.

**Claim 38 :**

Bristol, Choo, and Nelson do not explicitly disclose “wherein at least one Item is a member of an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said at least one Item.” Anglin discloses “wherein at least one Item is a member of an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said at least one Item.” (See page 3, paragraph [0028] “The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i.” This follows the request to delete the “group leader” which represents the group ID of the “storage

group” or the “Item Folder” as in referred to in the claim.) Bristol, Choo, Nelson, and Anglin are all directed to the same field of endeavor as applicant’s invention, as they are systems of file organization and management. It would have been obvious to a person of an ordinary skill in the art at the time the invention was disclosed to have applied Anglin’s disclosure above to the combination of Bristol, Choo, and Nelson for the purpose of providing a removal method for the item.

**Claim 39 :**

The combination of Bristol, Choo, Nelson, and Anglin disclose in Anglin “wherein the at least one Item is automatically deleted when it no longer belongs to any Item Folder.” (See page 3, paragraph [0029] “If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.”)

**Claim 41 :**

The combination of Bristol, Choo, Nelson, and Anglin disclose in Anglin “The method of claim 38 wherein the at least one Item is automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted.” (See page 3, paragraph [0028] “The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i.” This follows the request to delete the “group leader” which

represents the group ID of the “storage group” or the “Item Folder” as in referred to in the claim.)

**Claim 44 :**

Bristor, Choo, and Nelson do not explicitly disclose “wherein at least one Item is a member of an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said Item.” Anglin discloses “wherein the Item is a member of an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said Item.” (See page 3, paragraph [0028] “The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i.” This follows the request to delete the “group leader” which represents the group ID of the “storage group” or the “Item Folder” as in referred to in the claim.) Bristor, Choo, Nelson, and Anglin are all directed to the same field of endeavor as applicant’s invention, as they are systems of file organization and management. It would have been obvious to a person of an ordinary skill in the art at the time the invention was disclosed to have applied Anglin’s disclosure above to the combination of Bristor, Choo, and Nelson for the purpose of providing a removal method for the item.

**Claim 45 :**

The combination of Bristor, Choo, Nelson, and Anglin disclose in Anglin “The computer-readable medium of claim 44 wherein the at least one Item is automatically deleted when it no

longer belongs to any Item Folder” (See page 3, paragraph [0029] “If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.”).

**Claim 47 :**

The combination of Bristor, Choo, Nelson, and Anglin disclose in Anglin “The computer-readable medium of claim 44 wherein the at least one Item is automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted.” (See page 3, paragraph [0028] “The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i.” This follows the request to delete the “group leader” which represents the group ID of the “storage group” or the “Item Folder” as in referred to in the claim.)

**12. Claims 4, 6, 9, and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6018342 by David Bristor (hereafter Bristor) further in view of U.S. Patent Application Publication 20030009685 by Choo et. al. (hereafter Choo), U.S. Patent 7158962 by Kenneth Nelson (hereafter Nelson), and U.S. Patent Application Publication 2004/0073560 by Edwards (hereafter Edwards).**



**Claim 4 :**

The combination of Bristol, Choo, and Nelson do not explicitly disclose "wherein an Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder." On the other hand, Edwards teaches more explicitly an Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder). It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Bristol, Choo, and Nelson with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Bristol, Choo, and Nelson. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have an Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

**Claim 6 :**

Bristol, Choo, and Nelson do not explicitly disclose "said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder." However, Edwards teaches said Item, when it is a member of only one Item Folder and

said Item Folder is deleted, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] “The synchroniser can be set to ‘Recycle’ rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be ‘recycled’. This means it is not deleted immediately, but stored in an area where it can be retrieved if required.” In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Bristol, Choo, and Nelson with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Bristol, Choo, and Nelson. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.

**Claim 9 :**

Bristol, Choo, and Nelson do not explicitly teach “each said Item, when each no longer belongs to any Item Folder, automatically become members of a default Item Folder.” Nelson does disclose col. 5 lines 54-56, No folder exists matching these attributes. Consequently, the library server 25 creates a folder with the following attributes. However, Edwards more explicitly teaches “each said Item, when each no longer belongs to any Item Folder, automatically become members of a default Item Folder.” (See page 3, paragraph [0038] “The synchroniser can be set to ‘Recycle’ rather than delete files. This means that whenever the synchroniser is to over-write

or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Bristol, Choo, and Nelson with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Bristol, Choo, and Nelson. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have each said Item, when each no longer belongs to any Item Folder, automatically become members of a default Item Folder.

**Claim 11 :**

Bristol, Choo, and Nelson do not explicitly teach "each said Item, when each is a member of only one Item Folder and said Item Folder is deleted, automatically become members of a default Item Folder." Nelson does disclose col. 5 lines 54-56, No folder exists matching these attributes. Consequently, the library server 25 creates a folder with the following attributes. However, Edwards teaches "each said Item, when each is a member of only one Item Folder and said Item Folder is deleted, automatically become members of a default Item Folder." (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is

stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Bristor, Choo, and Nelson with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Bristor, Choo, and Nelson. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have each said Item, when each is a member of only one Item Folder and said Item Folder is deleted, automatically become members of a default Item Folder.

**13. Claims 13-20 rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6018342 by David Bristor (hereafter Bristor) further in view of U.S. Patent Application Publication 20030009685 by Choo et. al. (hereafter Choo), U.S. Patent 7158962 by Kenneth Nelson (hereafter Nelson), and U.S. Patent 6578046 by Chang et. al. (hereafter Chang).**

**Claim 13 :**

Bristor, Choo, and Nelson do not explicitly disclose “The computer system of claim 1 wherein a category is defined by an Item property.” Nelson discloses col. 4 lines 60-62, automatic linking rules for that item type. These rules are stored in a separate table in the library server. However, Chang more explicitly teaches “a Category is defined by an Item property.” (See column 13, lines 26-30 “In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them.” Simply by the Category

being formed as the result of a query, the Item property is necessarily what defined the Category, as the data must meet the Item property in order to be a result of the query.) It would have been obvious to one with ordinary skill in the art to combine the system as disclosed in Bristol, Choo, and Nelson with the disclosure of Chang because it would be logical to use the Item property as what defines the categories, especially in the case of a query. It is for this reason that one of ordinary skill in the art would have been motivated to have a Category is defined by an Item property.

**Claim 14 :**

Bristol, Choo, and Nelson do not explicitly disclose “wherein one of said plurality of Categories is defined by an Item Property and only an Item comprising the Item property for a specific Category from among said plurality of Categories can be a member of said specific Category.” The combination of Bristol, Choo, and Nelson disclose in Nelson col. 4 lines 60-62, automatic linking rules for that item type. These rules are stored in a separate table in the library server. However, Chang teaches more explicitly “wherein one of said plurality of Categories is defined by an Item Property and only an Item comprising the Item property for a specific Category from among said plurality of Categories can be a member of said specific Category.” (See column 13, lines 26-30 “In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them.” Simply by the Category being formed as the result of a query, the Item property is necessarily what defined the Category, as the data must meet the Item property in order to be a result of the query. Also, by the nature

of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property.) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in Bristol, Choo, and Nelson with the disclosure of Chang because it would be logical to use the Item property as what defines the categories, especially in the case of a query. It is for this reason that one of ordinary skill in the art would have been motivated to have one of said plurality of Categories is defined by an Item property, and only an Item comprising the Item property for a specific Category from among said plurality of Categories can be a member of said specific Category.

**Claim 15 :**

Bristol, Choo, and Nelson do not explicitly disclose “an Item comprising the Item property for one of said plurality of Categories is automatically a member of that one of said plurality of Categories.” However, Chang teaches “an Item comprising the Item property for one of said plurality of Categories is automatically a member of that one of said plurality of Categories.” (See column 13, lines 26-30 “In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them.” By the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property. Here, all of the results of the query are included in on the collection [referred to in the instant application as a category].) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in Bristol, Choo, and Nelson with the disclosure of Chang because keeping the results of the query, all of which

exhibit a relationship, is useful in that the query will not have to be run again. It is for this reason that one of ordinary skill in the art would have been motivated to include an Item comprising the Item property for one of said plurality of Categories is automatically a member of that one of said plurality of Categories.

**Claim 16 :**

Bristor, Choo, and Nelson do not explicitly disclose “an Item comprising one or more Item properties corresponding to one or more Categories of said plurality of Categories is automatically a member of each such Categories for said corresponding Item properties.” However, Chang teaches “an Item comprising one or more Item properties corresponding to one or more Categories of said plurality of Categories is automatically a member of each such Categories for said corresponding Item properties.” (See column 13, lines 26-30 “In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them.” By the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property. Here, all of the results of the query are included in on the collection [referred to in the instant application as a category].) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in Bristor, Choo, and Nelson with the disclosure of Chang because keeping the results of the query, all of which exhibit a relationship, is useful in that the query will not have to be run again. It is for this reason that one of ordinary skill in the art would have been motivated to include an Item comprising one or more Item properties corresponding to

one or more Categories of said plurality of Categories is automatically a member of each such Categories for said corresponding Item properties.

**Claim 17 :**

Bristor, Nelson, and Moraes do not explicitly disclose “wherein each of said plurality of categories is defined by an item property.” Nelson does disclose col. 4 lines 60-62, automatic linking rules for that item type. These rules are stored in a separate table in the library server. However, Chang more explicitly teaches “each of said plurality of Categories is defined by an Item property.” (See column 13, lines 26-30 “In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them.” Simply by the Category being formed as the result of a query, the Item property is necessarily what defined the Category, as the data must meet the Item property in order to be a result of the query.) It would have been obvious to one with ordinary skill in the art to combine the system as disclosed in Bristor, Choo, and Nelson with the disclosure of Chang because it would be logical to use the Item property as what defines the categories, especially in the case of a query. It is for this reason that one of ordinary skill in the art would have been motivated to have each of said plurality of Categories is defined by an Item property.

**Claim 18 :**

Bristor, Choo, and Nelson do not explicitly teach “each of said plurality of Categories is defined by an Item property, and only Items comprising the Item property for a specific Category from



among said plurality of Categories can be members of said specific Category.” However, Chang teaches “each of said plurality of Categories is defined by an Item property, and only Items comprising the Item property for a specific Category from among said plurality of Categories can be members of said specific Category.” (See column 13, lines 26-30 “In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them.” Simply by the Category being formed as the result of a query, the Item property is necessarily what defined the Category, as the data must meet the Item property in order to be a result of the query. Also, by the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property.) It would have been obvious to one with ordinary skill in the art to combine the system as disclosed in Bristol, Choo, and Nelson with the disclosure of Chang because it would be logical to use the Item property as what defines the categories, especially in the case of a query. It is for this reason that one of ordinary skill in the art would have been motivated to have each of said plurality of Categories is defined by an Item property, and only Items comprising the Item property for a specific Category from among said plurality of Categories can be members of said specific Category.

**Claim 19 :**

Bristol, Choo, and Nelson do not explicitly teach “each Item comprising the Item property for one of said plurality of Categories are automatically members of that one of said plurality of Categories.” However, Chang teaches “each Item comprising the Item property for one of said

plurality of Categories are automatically members of that one of said plurality of Categories.” (See column 13, lines 26-30 “In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them.” By the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property. Here, all of the results of the query are included in on the collection [referred to in the instant application as a category].) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in Bristol, Choo, and Nelson with the disclosure of Chang because keeping the results of the query, all of which exhibit a relationship, is useful in that the query will not have to be run again. It is for this reason that one of ordinary skill in the art would have been motivated to include each Item comprising the Item property for one of said plurality of Categories are automatically members of that one of said plurality of Categories.

**Claim 20 :**

Bristol, Choo, and Nelson do not explicitly teach “all Items comprising one or more Item properties corresponding to one or more Categories of said plurality of Categories are automatically members of all such Categories for said corresponding Item properties.” However, Chang teaches “all Items comprising one or more Item properties corresponding to one or more Categories of said plurality of Categories are automatically members of all such Categories for said corresponding Item properties.” (See column 13, lines 26-30 “In the preferred embodiment, a FederatedCollection allows an application program to process data objects

resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them.” By the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property. Here, all of the results of the query are included in on the collection [referred to in the instant application as a category].) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in Bristor, Choo, and Nelson with the disclosure of Chang because keeping the results of the query, all of which exhibit a relationship, is useful in that the query will not have to be run again. It is for this reason that one of ordinary skill in the art would have been motivated to include all Items comprising one or more Item properties corresponding to one or more Categories of said plurality of Categories are automatically members of all such Categories for said corresponding Item properties.

**14. Claim 23 is rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6018342 by David Bristor (hereafter Bristor) further in view of U.S. Patent Application Publication 20030009685 by Choo et. al. (hereafter Choo), U.S. Patent 7158962 by Kenneth Nelson (hereafter Nelson), and U.S. Patent 6438545 by Beauregard et. al. (hereafter Beauregard)**

**Claim 23 :**

Bristor, Choo, and Nelson do not explicitly disclose “a virtual machine manager”. However, Beauregard teaches said Item is a fundamental unit of information manipulated by a virtual machine manager. (See column 13, lines 12-16 “This broad I/O capability can be provided under the Virtual Machine Manager (VMM) that is available under Win32. The VMM is an

extensible operating system whose core and standard components are provided by Microsoft Corporation.”) Because of the advantages provided by VMM as taught in Beauregard, such as the broad I/O capability, it would have been obvious to one with ordinary skill in the art to combine the VMM of Beauregard with the teaching of Bristor, Choo, and Nelson. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item is a fundamental unit of information manipulated by a virtual machine manager.

**15. Claims 40, 42, 46, and 48 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6018342 by David Bristor (hereafter Bristor) further in view of U.S. Patent Application Publication 20030009685 by Choo et. al. (hereafter Choo), U.S. Patent 7158962 by Kenneth Nelson (hereafter Nelson), U.S. Patent Application Publication 2004/0199521 by Anglin et. al (hereafter Anglin), and U.S. Patent Application Publication 2004/0073560 by Edwards (hereafter Edwards).**

**Claim 40 :**

Bristor, Choo, Nelson, and Anglin do not explicitly teach “said at least one Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.” However, Edwards teaches said Item, “when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.” (See page 3, paragraph [0038] “The synchroniser can be set to ‘Recycle’ rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be ‘recycled’. This means it is not deleted immediately, but stored in an area where it can be retrieved if required.” In the instant application, the area in which the item is stored is called the

default Item Folder). It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Bristol, Choo, Nelson, and Anglin with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Bristol, Choo, Nelson, and Anglin. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

**Claim 42 :**

Bristol, Choo, Nelson, and Anglin do not explicitly teach “said at least one Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.” However, Edwards teaches “said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.” (See page 3, paragraph [0038] “The synchroniser can be set to ‘Recycle’ rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be ‘recycled’. This means it is not deleted immediately, but stored in an area where it can be retrieved if required.” In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Bristol, Choo, Nelson, and Anglin with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Bristol, Choo, Nelson, and

Anglin. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.

**Claim 46 :**

Bristor, Choo, Nelson, and Anglin do not explicitly disclose “said at least one Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.” However, Edwards teaches “said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.” (See page 3, paragraph [0038] “The synchroniser can be set to ‘Recycle’ rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be ‘recycled’. This means it is not deleted immediately, but stored in an area where it can be retrieved if required.” In the instant application, the area in which the item is stored is called the default Item Folder) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Bristor, Choo, Nelson, and Anglin with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Bristor, Choo, Nelson, and Anglin. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been

motivated to have said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

**Claim 48 :**

The combination of Bristor, Choo, Nelson, and Anglin do not explicitly disclose “wherein said at least one item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.” However, Edwards teaches “said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.” (See page 3, paragraph [0038] “The synchroniser can be set to ‘Recycle’ rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be ‘recycled’. This means it is not deleted immediately, but stored in an area where it can be retrieved if required.” In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Bristor, Choo, Nelson, and Anglin with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Bristor, Choo, Nelson, and Anglin. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.

***Response to Arguments***

16. Applicant's arguments filed 6/16/2010 have been fully considered but they are not persuasive. Applicant's assert the following with regard to the cited references:

A. That Bristor and Nelson fails to disclose the database management program that is part of an operating system and controls access to a file system. That just because a program runs on a computer system that also has an operating system does not mean that the program is a part of the operating system.

In response, this is respectfully disagreed, a program running on an operating system is a part of the operating system. Furthermore the limitation was stated to be disclosed by Choo.

B. That Choo does not disclose the database management program that is part of an operating system and controls access to a file system.

In response this is disagreed. See figure 2-3. Accordingly, the computer readable storage medium (figure 3) including instructions for an operating system (figure 2 element 200, system) including instructions for a database management program (figure 2 element 203, rules database), the instructions for the database management program (figure 2 element 203, rules database) configured to control (0014 line 15-16, determine whether the compartment associated with process 201 is permitted access) user mode application (figure 2 element 201, process



executes code in user-space) access to a file system (0014 line 16, permitted access to the particular resource)

C. That the database of Choo fails to control user mode application access to the file system.

In response this is disagreed. The rules database of Choo does control user mode application access to the file system as shown in the above rejection. Choo's database (rules database) controls (permits) user mode application access (process permitted access) to the file system (resources).

D. Applicant's further assert the new limitations in claim 1, 37, and 43 which all similarly recite "the instructions for the database management program further including instructions for receiving receive read/write requests from the user mode applications for Items via one or more functions of an operating system application program interface; and the instructions for the database management program further including instructions for deserializing files storing the file data for the Items into Items and return the Items to the user mode applications."

In response, please see the above rejections of claim 1, 37, and 43.

All assertions have been traversed, the rejection is therefore maintained.

***Conclusion***

17. The prior art made of record listed on pto-892 and not relied, if any, upon is considered pertinent to applicant's disclosure.

***Contact Information***

18. Any inquiry concerning this communication or earlier communications from the examiner should be directed to MICHAEL PHAM whose telephone number is (571)272-3924. The examiner can normally be reached on 9am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cottingham can be reached on 571-272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/M. P./  
Examiner, Art Unit 2167

/John R. Cottingham/  
Supervisory Patent Examiner, Art Unit  
2167

